Remarque 1 *Ce sujet propose une partie de programmation, en principe sous Maple ou Mathematica. Afin de coller au changement de programme, on utilisera le langage Python.*

I Étude d'un endomorphisme

I.A -

I.A.1) On obtient facilement : $M = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$

I.A.2) La cinquième ligne étant nulle, le déterminant est nul.

I.A.3) M n'est donc pas inversible, le rang de f ou de M n'est donc pas 6, il est donc inférieur ou égal à 5.

I.A.4) Les vecteurs $\left(\overrightarrow{e_1}, \overrightarrow{e_2}, \overrightarrow{e_3}, \overrightarrow{e_4}, \overrightarrow{e_6}\right)$ sont invariants, ils sont donc dans l'image de f.

I.A.5) Les E_k sont de dimension 5, engendrés par 5 vecteurs formant une famille libre.

I.A.6) $\mathrm{Im}(f)$ est de dimension inférieure ou égale à 5 et contient E_5 qui est de dimension 5.

Donc l'image de f est de dimension 5 et les vecteurs $\left(\overrightarrow{e_1}, \overrightarrow{e_2}, \overrightarrow{e_3}, \overrightarrow{e_4}, \overrightarrow{e_6}\right)$ en forment une base !

On a bien : $\mathrm{Im}(f) = E_5$, et, par théorème du rang, le noyau de f est de dimension 1.

I.B - Le noyau est de dimension 1, un seul vecteur non nul suffit donc.

On cherche d'abord le noyau de M.

On écrit : $MX = 0$, avec X le vecteur colonne de coordonnées x_i.

On obtient le système : $\begin{cases} x_1 + x_5 = 0 \\ x_2 - x_5 = 0 \\ x_3 = 0 \\ x_4 = 0 \\ x_5 + x_6 = 0 \end{cases}$

On prend $x_5 = -1$, on obtient le vecteur $(1, -1, 0, 0, -1, 1)$.

Un vecteur du noyau de f est donc : $\overrightarrow{u} = \overrightarrow{e_1} - \overrightarrow{e_2} - \overrightarrow{e_5} + \overrightarrow{e_6}$.

I.C -

I.C.1) $P_M(\lambda) = \begin{vmatrix} 1-\lambda & 0 & 0 & 0 & 1 & 0 \\ 0 & 1-\lambda & 0 & 0 & -1 & 0 \\ 0 & 0 & 1-\lambda & 0 & 0 & 0 \\ 0 & 0 & 0 & 1-\lambda & 0 & 0 \\ 0 & 0 & 0 & 0 & -\lambda & 0 \\ 0 & 0 & 0 & 0 & 1 & 1-\lambda \end{vmatrix} = (1-\lambda)^4 \begin{vmatrix} -\lambda & 0 \\ 1 & 1-\lambda \end{vmatrix} = -\lambda(1-\lambda)^5.$

I.C.2) On a donc : 1 est valeur propre d'ordre 5, et 0 est valeur propre simple.

I.C.3) Les vecteurs de la base de E_5 sont invariants, donc propre pour la valeur propre 1.

Donc, E_5 est inclus dans le sous-espace propre associé à la valeur propre 1, qui est de dimension au plus 5.

E_5 est donc le sous-espace propre associé à la valeur propre 1.

D'autre part, le noyau de f, s'il n'est pas réduit au vecteur nul, est toujours le sous-espace propre associé à la valeur propre 0.

I.C.4) La somme des dimensions des sous-espaces propres est 6, donc f est diagonalisable.
Une base de vecteurs propres est : $\left(\overrightarrow{e_1}, \overrightarrow{e_2}, \overrightarrow{e_3}, \overrightarrow{e_4}, \overrightarrow{e_6}, \overrightarrow{u}\right)$.

I.D -

I.D.1) Prenons un vecteur $\overrightarrow{v}$ quelconque :
$f \circ f\left(\overrightarrow{v}\right) = f\left(f\left(\overrightarrow{v}\right)\right) = f\left(\overrightarrow{v}\right)$, car $f\left(\overrightarrow{v}\right) \in E_5$, et est donc propre pour la valeur propre 1.
f est donc un projecteur, projection sur l'image, E_5, parallèlement au noyau, $\mathbb{R}\overrightarrow{u}$.

I.D.2) $\overrightarrow{u}$ n'est pas orthogonal à $\overrightarrow{e_1}$, qui appartient à E_5, ce projecteur n'est donc pas orthogonal.

I.D.3) $M^2 = M$ puisque $f \circ f = f$!

I.E -

I.E.1) On a bien $\overrightarrow{b} \in E_5$, donc $f\left(\overrightarrow{b}\right) = \overrightarrow{b}$.
$\overrightarrow{b}$ est bien une solution de $f\left(\overrightarrow{x}\right) = \overrightarrow{b}$.

I.E.2) f est linéaire et les solutions de $f\left(\overrightarrow{x}\right) = \overrightarrow{0}$ sont les $\lambda\overrightarrow{u}$.
La solution générale de $f\left(\overrightarrow{x}\right) = \overrightarrow{b}$ est : $\overrightarrow{x} = \overrightarrow{b} + \lambda\overrightarrow{u}$.

I.E.3) On a donc : $\overrightarrow{x} = (4+\lambda, 4-\lambda, 4, 1, -\lambda, 3+\lambda)$ avec $\lambda \neq 0$ puisque $\overrightarrow{x} \neq \overrightarrow{b}$.
Pour que les coordonnées soient positives, il faut et il suffit que $\lambda \leqslant 0$ et $\lambda \geqslant -3$.
Alors, les 4 premières coordonnées de $\overrightarrow{x}$ sont non nulles, seule la dernière peut être nulle compte tenu de $\lambda \neq 0$, donc $\lambda = -3$ et $\overrightarrow{c} = (1, 7, 4, 1, 3, 0)$ est bien cette solution.

II Solutions approchées

II.A -

II.A.1) ${}^t\!AAX = {}^t\!AY_0 \Leftrightarrow {}^t\!A(AX - Y_0) = 0$
$AZ \cdot (AX - Y_0) = {}^tZ\,{}^t\!A(AX - Y_0)$
On a donc bien : ${}^t\!AAX = {}^t\!AY_0 \Rightarrow AZ \cdot (AX - Y_0) = 0$.
Réciproquement, si $AZ \cdot (AX - Y_0) = 0$, ${}^tZ\,{}^t\!A(AX - Y_0) = 0$.
En prenant Z le k-ème vecteur de la base, on annule ainsi la k-ème coordonnée de ${}^t\!A(AX - Y_0)$.
Comme on peut le faire pour tous les vecteurs, donc tous les vecteurs de la base, on obtient : ${}^t\!A(AX - Y_0) = 0$.
On a bien montré que : ${}^t\!AAX = {}^t\!AY_0$ si et seulement si $AX - Y_0$ est orthogonal à tout vecteur de la forme AZ.

II.A.2) Cela revient à dire que : ${}^t\!AAX = {}^t\!AY_0$ si et seulement si $AX - Y_0$ est orthogonal à tout vecteur de l'image de A.
Comme AX appartient à l'image de A, cela revient à dire que AX est le projeté orthogonal de Y_0 sur l'image de A.

II.A.3) Comme X décrit $\mathbb{R}^6$, AX décrit l'image de A.
Il existe donc un vecteur X tel que AX est le projeté orthogonal de Y_0 sur l'image de A.

II.B -

II.B.1) ${}^tB = {}^t\left({}^t\!AA\right) = {}^t\!AA = B$, B est bien symétrique.

II.B.2) On a : ${}^tXBX = {}^tX\,{}^t\!AAX = AX \cdot AX = \|AX\|^2 \geqslant 0$.
Mais on a aussi : ${}^tXBX = \lambda\,{}^tXX = \lambda\,\|X\|^2$.
Comme $\|X\|^2 > 0$, puisque X est un vecteur propre, on obtient : $\lambda \geqslant 0$.

II.B.3) Comme B est symétrique réelle, elle est diagonalisable dans une base orthonormale de vecteurs propres, c'est à dire avec une matrice de passage orthogonale.

Comme B est de rang 5, elle a $\lambda = 0$ qui est valeur propre simple, les autres valeurs propres sont non nulles, donc strictement positives.

En prenant le vecteur propre associé à la valeur propre 0 de la base orthogonale de vecteurs propres en dernier, on a les conditions demandées par l'énoncé.

II.B.4) ${}^tAAX = {}^tAY_0 \Leftrightarrow BX = {}^tAY_0 \Leftrightarrow {}^tPDPX = {}^tAY_0 \Leftrightarrow DPX = P\,{}^tAY_0$

En posant $X' = PX$ et on calcule $Y_0' = P\,{}^tAY_0$, l'équation revient alors à $DX' = Y_0'$.

X existe, donc X' aussi, les dernières coordonnées de X' et Y_0' sont donc nulles, et le système se ramène à : $\begin{cases} \lambda_1 x_1' = y_1' \\ \lambda_2 x_2' = y_2' \\ \lambda_3 x_3' = y_3' \\ \lambda_4 x_4' = y_4' \\ \lambda_5 x_5' = y_5' \end{cases}$

X' se trouve donc facilement, et il faut calculer $X = {}^tPX'$ pour terminer.

***II.C* -** X_0 est une solution de ${}^tAAX = {}^tAY_0$.

La distance de AX à Y_0 est minimale au sens des moindres carrés si et seulement si AX et la projection orthogonale de Y_0 sur l'espace vectoriel décrit par les AX, c'est à dire l'image de A.

On a ${}^tAAX_0 = {}^tAY_0$, donc $X_0\,{}^tAAX_0 = X_0\,{}^tAY_0$, ou encore : $\|AX_0\| = AX_0 \cdot Y_0$.

Ce qui prouve que AX_0 est le projeté orthogonal de Y_0 sur l'image de A.

Donc : $X_0 \in \mathscr{P}$.

III Applications linéaires et manipulation de cruches

***III.A* -**

III.A.1) On calcule : $MX = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix} = \begin{pmatrix} x_1 + x_5 \\ x_2 - x_5 \\ x_3 \\ x_4 \\ 0 \\ x_5 + x_6 \end{pmatrix}$

III.A.2) $v(3,1)$ consiste à vider la cruche 3 dans la cruche 1.

La cruche 3 se retrouve vide.

y_5 est donc nul, et y_6 vaut le volume d'air qu'il y avait, x_6, plus le volume d'eau qu'il y avait dans la cruche 3, x_5

La cruche 1 se retrouve avec l'eau qu'elle avait, x_1, plus l'eau de la cruche 3, x_5.

Donc : $y_1 = x_1 + x_5$.

La quantité d'air y a diminué de l'eau qu'on a ajouté : $y_2 = x_2 - x_5$.

Enfin, on n'a pas touché à la cruche 2 : $y_3 = x_3$ et $y_4 = x_4$.

C'est bien la manipulation qui correspond à la matrice M.

***III.B* -** On observe que C_k correspond aux indices $2k-1$, pour le volume d'eau, et $2k$, pour le volume d'air.

III.B.1) On vide C_i, pour l'eau : $y_{2i-1} = 0$, et pour l'air : $y_{2i} = x_{2i} + x_{2i-1}$.

On la vide dans C_j, pour l'eau : $y_{2j-1} = x_{2j-1} + x_{2i-1}$, et pour l'air : $y_{2j} = x_{2j} - x_{2i-1}$.

On ne touche pas aux autres cruches, donc : $y_k = x_k$, pour les valeurs non référencées ci-dessus.

III.B.2) On remplit C_j avec C_i, La quantité d'eau qu'on déplace est donc le volume d'air de C_j : x_{2j} !
Pour C_j : $y_{2j-1} = x_{2j-1} + x_{2j}$ et $y_{2j} = 0$.
Pour C_i : $y_{2i-1} = x_{2i-1} - x_{2j}$ et $y_{2i} = x_{2i} + x_{2j}$.
Pour les autres cruches, et donc les autres indices : $y_k = x_k$.
En fait, on inverse les rôles de i et j, ainsi que des indices $2k$ et $2k-1$.

III.C -

III.C.1) E_k est de dimension 5, c'est donc un hyperplan de $\mathbb{R}^6$.

III.C.2) Un équation de cet hyperplan est bien sûr : $x_k = 0$.

III.D -

III.D.1) Un vecteur de E_{2i-1} vérifie $x_{2i-1} = 0$, il n'y a pas d'eau dans C_i : la vider consiste à ne rien faire.
On peut aussi le vérifier sur les coordonnées.

III.D.2) Un vecteur de E_{2j} vérifie $x_{2j} = 0$, il n'y a pas d'air dans C_j : la remplir consiste à ne rien faire.
Ce qui se vérifie aussi sur les coordonnées...

III.D.3) Quand on applique $M_{i,j}$, le résultat vérifie toujours $y_{2i-1} = 0$: la $(2i-1)$-ème ligne de $M_{i,j}$ est nulle.
Pour la matrice correspondant à $P_{i,j}$, on aura toujours $y_{2j} = 0$: la $2j$-ème ligne de cette matrice est nulle.

III.D.4) $V_{i,j}$ et $V_{i,j}$ ont tous les deux un sous espace propre de dimension 5 pour la valeur propre 1.
Leur rang est donc au moins 5.
Mais leur rang ne peut être 6 car leur déterminant est nul, leur matrice ayant une ligne de zéros.
Leur rang est donc 5. Elles sont diagonalisables car elles ont un sous espace propre de dimension 5 associé à la valeur propre 1, et un sous espace propre associé à la valeur propre 0, de dimension au moins 1, et ici donc, de dimension 1.

IV Le problème des trois cruches

IV.A -

IV.A.1) À l'état initial, on a $x_1 = 1$ et $x_3 = x_5 = 0$. Compte tenu de la capacité de chaque cruche, $x_2 = 0$, $x_4 = 5$ et $x_6 = 3$.
Cela correspond bien au vecteur $\overrightarrow{a}$.
À l'état final, on a $x_1 = x_3 = 4$ et $x_5 = 0$. Toujours compte tenu de la capacité de chaque cruche, $x_2 = 4$, $x_4 = 1$ et $x_6 = 3$.
Cela correspond bien au vecteur $\overrightarrow{b}$.

IV.A.2) Le vecteur $\overrightarrow{c}$ correspond à un remplissage respectif de chaque cruche de 1 litre, 4 litres et 3 litres.
Si on vide C_3 dans C_1, on aura bien toujours respectivement 4 litres, 4 litres, et la troisième cruche vide.
Ce qui est l'état final demandé.

IV.A.3) Si on remplit la cruche C_j, alors on aura $y_{2j} = 0$.
Si on vide la cruche C_i, alors on aura $y_{2i-1} = 0$.
On aura toujours une coordonnée nulle au moins, les autres sont naturellement positives à moins de faire une manipulation physiquement impossible !

IV.B -

IV.B.1) Le vecteur $\overrightarrow{c}$ correspond à un remplissage respectif de chaque cruche de 1 litre, 4 litres et 3 litres.
On peut faire :

- $V_{1,2} = P_{1,2}$ qui laisse 2 cruches pleines, c'est non privilégié.
- $V_{3,1}$ qui donne l'état final.
- $V_{2,1}$ qui donne C_2 vide et C_3 pleine, non privilégié.
- $P_{3,2}$ qui donne 1 litre, 5 litres et 2 litres qui donne le vecteur : $(1, 7, 5, 0, 2, 1) = \overrightarrow{d}$.
- et rien d'autre !

IV.B.2) Il suffit de remplir C_3 avec C_2 pour repasser de $\overrightarrow{d}$ à $\overrightarrow{c}$...

IV.C - Le vecteur $\overrightarrow{a}$ correspond à un remplissage respectif de chaque cruche de 8 litre, 0 litres et 0 litres.

Le vecteur $\overrightarrow{h}$ correspond à un remplissage respectif de chaque cruche de 3 litres, 5 litres et 0 litres.

Il suffit de remplir C_2 avec C_1 pour passer de l'un à l'autre.

IV.D - Successivement :

- Remplir C_2 avec C_1
- Remplir C_3 avec C_2
- Vider C_3 dans C_1
- Vider C_2 dans C_3
- Remplir C_2 avec C_1
- Remplir C_3 avec C_2
- Vider C_3 dans C_1

IV.E -

IV.E.1) Les volumes d'eau sont toujours entiers, il est donc impossible de répartir en parts égales 8 litres dans 3 cruches.

IV.E.2) La répartition 4 litres, 2 litres et 2 litres est impossible puisqu'on a toujours une cruche vide ou une cruche pleine.

IV.F -

IV.F.1) On a $Y_0 = (8/3, 16/3, 8/3, 7/3, 8/3, 1/3)$ en écrivant le vecteur en ligne par commodité.

IV.F.2) La distance sera la plus courte en projetant sur E_6,
le projeté orthogonal sera $(8/3, 16/3, 8/3, 7/3, 8/3, 0)$, et la distance sera égale à $1/3$.

V Utilisation du logiciel de calcul

On utilise ici trois listes appelées `u,v,w` pour mémoriser les états successifs de remplissage des trois cruches.
De même, le paramètre m est inutile et supprimé...
Enfin, l'affichage n'aura pas tout à fait la forme demandée, toujours pour simplifier l'écriture des procédures...

V.A - On ne prendra pas ici trois paramètres en entrée, mais une liste de trois paramètres appelée `A`, pour simplifier la solution finale.

```
def figure(A):
    for j in range(len(u)):
        if u[j]==A[0] and v[j]==A[1] and w[j]==A[2]:
            return True
    return False
```

V.B - On va ici procéder plus directement que par les procédures demandées.

Initialisation

```
u=[8];v=[0];w=[0] # Situation de départ
Cible=[4,4,0]     # Cible
Vmax=[8,5,3]      # Contenance des cruches
```

Mise à jour des états après une opération

```
def Accoler(A):
    u.append(A[0])
    v.append(A[1])
    w.append(A[2])
```

Affichage de la situation

```
def Affiche_etat():
    print('m=',len(u))
    print(u)
    print(v)
    print(w)
```

Remplissage si possible

```
def Remplir(i,j):
    A=[u[-1],v[-1],w[-1]]
    if A[i]!=0 and A[i]+A[j]>=Vmax[j]:    # Possible à priori
        A[i]=A[i]+A[j]-Vmax[j]            # Attention à l'ordre
        A[j]=Vmax[j]                      # de ces deux lignes !
        if not figure(A):                 # Nouvel état
            Accoler(A)
            Affiche_etat()
```

Vidage si possible

```
def Vider(i,j):
    A=[u[-1],v[-1],w[-1]]
    if A[i]!=0 and A[i]+A[j]<=Vmax[j]:    # Possible à priori
        A[j]=A[i]+A[j]                    # Attention à l'ordre
        A[i]=0                            # de ces deux lignes !
        if not figure(A):                 # Nouvel état
            Accoler(A)
            Affiche_etat()
```

V.C - Tout le travail est fait, ou presque !

Boucle finale

```
while not figure(Cible):
    for i in range(3):
        for j in range(3):
            if i!=j:
                Remplir(i,j)
                Vider(i,j)
```

V.D - Passer m à 8 et ajouter la colonne $\begin{bmatrix}4\\4\\0\end{bmatrix}$ au bout de T.

V.E - Sans surprise, c'est exactement ce qu'on a déjà décrit !...